

Practical 6

Aim: To apply the concept of Aggregating Data using Group functions.

(1) List total deposit of customer having account date after 1-jan-96.

Query: select sum(amount) from deposit where a_date>'01-jan-96'

Output:

User: 22DCE006

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select sum(amount) from deposit where a_date>'01-jan-96'
```

Results Explain Describe Saved SQL History

SUM(AMOUNT)
39500

1 rows returned in 0.01 seconds [CSV Export](#)

(2) List total deposit of customers living in city Nagpur.

Query: select sum(amount) as "SUM" from deposit where bname='nagpur'

Output:

User: 22DCE006

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select sum(amount) as "SUM" from deposit where bname='nagpur'
```

Results Explain Describe Saved SQL History

SUM
-

1 rows returned in 0.00 seconds [CSV Export](#)

(3) List maximum deposit of customers living in bombay.

Query:select max(amount) as "MAX" from deposit where bname='bombay'

Output:

User: 22DCE006

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select max(amount) as "MAX" from deposit where bname='bombay'
```

Results Explain Describe Saved SQL History

MAX
-

1 rows returned in 0.00 seconds [CSV Export](#)

(4) Display the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number.

Query: select round(max(emp_sal)) as "Maximum Salary" , round(min(emp_sal)) as "Minimum Salary" , round(sum(emp_sal)) as "Total Salary" , round(avg(emp_sal)) as "Average Salary" from employee

Output:

User: 22DCE006

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼ Save Run

```
select round(max(emp_sal)) as "Maximum Salary" , round(min(emp_sal)) as "Minimum Salary" , round(sum(emp_sal)) as "Total Salary" , round(avg(emp_sal)) as "Average Salary" from employee
```

Results Explain Describe Saved SQL History

Maximum Salary	Minimum Salary	Total Salary	Average Salary
5000	800	16925	2418

1 rows returned in 0.00 seconds [CSV Export](#)

(5) Write a query that displays the difference between the highest and lowest salaries. Label the column DIFFERENCE.

Query: select round(max_sal) as "Highest Salary" , round(min_sal) as "Lowest Salary" , round((max_sal)-(min_sal))as "Difference" from job

Output:

User: 22DCE006

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
select round(max_sal) as "Highest Salary" , round(min_sal) as "Lowest Salary" , round((max_sal)-(min_sal))as "Difference" from job
```

Results Explain Describe Saved SQL History

Highest Salary	Lowest Salary	Difference
10000	4000	6000
15000	9000	6000
12000	8200	3800
9000	4200	4800
17000	6000	11000
3000	1500	1500

6 rows returned in 0.01 seconds [CSV Export](#)

(6) Create a query that will display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998

Query: select count(emp_name) as "Count" from employee where to_char(hiredate,'yy') in('95','96','97','98')

Output:

User: 22DCE006

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
select count(emp_name) as "Count" from employee where to_char(hiredate,'yy') in('95','96','97','98')
```

Results Explain Describe Saved SQL History

Count
7

1 rows returned in 0.00 seconds [CSV Export](#)

(7) Find the average salaries for each department without displaying the respective department numbers.

Query: select dept_no, ROUND(AVG(emp_sal)) from employee GROUP BY dept_no

Output:

User: 22DCE006

Home > SQL > **SQL Commands**

☒ Autocommit Display 10

```
SELECT DEPT_NO, ROUND(AVG(emp_sal)) FROM EMPLOYEE GROUP BY DEPT_NO
```

Results Explain Describe Saved SQL History

DEPT_NO	ROUND(AVG(EMP_SAL))
25	1600
30	2975
20	1100
10	2813

4 rows returned in 0.01 seconds [CSV Export](#)

(8) Write a query to display the total salary being paid to each job title, within each department.

Query: select dept_name, SUM(emp_sal) from employee GROUP BY dept_name

Output:

User: 22DCE006

Home > SQL > **SQL Commands**

☒ Autocommit Display 10

```
select dept_name, SUM(emp_sal) from employee GROUP BY dept_name
```

Results Explain Describe Saved SQL History

DEPT_NAME	SUM(EMP_SAL)
big data analytics	7450
artificial intelligence	2975
machine learning	1900
virtual reality	3000
data science	1600

5 rows returned in 0.00 seconds [CSV Export](#)

(9) Find the average salaries > 2000 for each department without displaying the respective department numbers.

Query: select ROUND(AVG(emp_sal)) from employee GROUP BY dept_no HAVING AVG(EMP_SAL)>2000

Output:

User: 22DCE006

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select ROUND(AVG(emp_sal)) from employee GROUP BY dept_no HAVING AVG(EMP_SAL)>2000
```

Results Explain Describe Saved SQL History

ROUND(AVG(EMP_SAL))
2975
2813

2 rows returned in 0.00 seconds [CSV Export](#)

(10) Display the job and total salary for each job with a total salary amount exceeding 3000 and sorts the list by the total salary.

Query: select job_id,SUM(emp_sal+3000)"EXTENDED SALARY " from employee GROUP BY job_id order by SUM(emp_sal+3000)

Output:

User: 22DCE006

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select job_id,SUM(emp_sal+3000)"EXTENDED SALARY " from employee GROUP BY job_id order by SUM(emp_sal+3000)
```

Results Explain Describe Saved SQL History

JOB_ID	EXTENDED SALARY
fi_mgr	3800
mk_mgr	4100
lec	4600
fi_acc	5450
it_prog	5975
comp_op	14000

6 rows returned in 0.01 seconds [CSV Export](#)

(11) List the branches having sum of deposit more than 5000 and located in city bombay.

Query: select bname from deposit group by bname having sum(amount)>5000

Output:

User: 22DCE006

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▾

```
select bname from deposit group by bname having sum(amount)>5000
```

Results Explain Describe Saved SQL History

BNAME
andheri
borivali
villepark
dadar

4 rows returned in 0.00 seconds [CSV Export](#)

Conclusion: From this practical I learned about different aggregate functions and other SQL group functions.

Staff Signature:

Grade:

Remarks by the Staff: